Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs
Generate OACS

Search Results - Record(s) 1 through 50 of 63 returned.

□ 1. Document ID: US 6678882 B1

L6: Entry 1 of 63 File: USPT Jan 13, 2004

US-PAT-NO: 6678882
DOCUMENT-IDENTIFIER: US 6678882 B1

TITLE: Collaborative model for software systems with synchronization submodel with merge feature, automatic conflict resolution and isolation of potential changes for reuse

DATE-ISSUED: January 13, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Hurley; William D.	Louisville	CO		
Habermehl; Kyle D.	Niwot	CO		

US-CL-CURRENT: 717/121; 717/122, 719/316

ABSTRACT:

A collaboration model is provided for constructing evolutionary collaborative applications deployed in network environments, such as the World Wide Web, or wireless environments, which supports introduction of new object types into the deployed system. The collaboration model is arranged as a set of submodels. The highest-level control resides in a synchronization submodel. A merge submodel defines mechanisms for merging changes, detecting conflicts, and resolving conflicts whenever possible. A change submodel is operative to define allowable changes for shared objects. A replication model is arranged to provide replication functionality, and define how multiple versions of an object are presented. A distribution submodel provides distribution functionality, and a type submodel provides a programming language-independent representational scheme for the overall collaboration model. The synchronization submodel, the merge submodel, and the replication submodel are arranged to use constructs defined by the change submodel.

25 Claims, 9 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 6

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Drawi D

□ 2. Document ID: US 6662357 B1

L6: Entry 2 of 63   File: USPT   Dec 9, 2003

US-PAT-NO: 6662357
DOCUMENT-IDENTIFIER: US 6662357 B1

TITLE: Managing information in an integrated development architecture framework

DATE-ISSUED: December 9, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bowman-Amuah; Michel K.	Colorado Springs	CO		

US-CL-CURRENT: 717/120

ABSTRACT:

A system, method, and article of manufacture are provided for managing information in a development architecture framework. Common information that is used by a plurality of components of a system is allowed to be accessed in a single, shared repository. Unique information that is unique to the components of the system is stored in corresponding designated folders. Media content communicated in the system is managed based on metadata thereof.

18 Claims, 14 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 14

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

□ 3. Document ID: US 6640249 B1

L6: Entry 3 of 63   File: USPT   Oct 28, 2003

US-PAT-NO: 6640249
DOCUMENT-IDENTIFIER: US 6640249 B1

TITLE: Presentation services patterns in a netcentric environment

DATE-ISSUED: October 28, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bowman-Amuah; Michel K.	Colorado Springs	CO		

US-CL-CURRENT: 709/228; 719/315

ABSTRACT:

A system, method and article of manufacture are provided for implementing presentation services patterns. Non-presentation logic executed on a client is assigned to an activity for allowing reuse of the non-presentation logic across multiple, volatile user interfaces. A view is assigned to the activity. Validation rules are also structured for validating user data across the multiple user interfaces.

15 Claims, 195 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 123

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

□ 4. Document ID: US 6618765 B1

L6: Entry 4 of 63 File: USPT Sep 9, 2003

US-PAT-NO: 6618765
DOCUMENT-IDENTIFIER: US 6618765 B1

TITLE: Method of modifying a protocol between distributed objects

DATE-ISSUED: September 9, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Banctel; Fabrice	Gif-sur-Yvette			FR
Pietre; Armel	Paris			FR

US-CL-CURRENT: 719/315

ABSTRACT:

A method of implementing a personalized call-back protocol between an object X of a client process and an object S of a server process in an environment based on an ORB distributed object manager allocates a connection point CP3 of the server object S to a client object X of a call-back service offered by the server object. The connection point includes a marshaling interface IMarshal for implementing an object-object protocol between the client object X and the connection point CP3 by means of a personalized pair of representative elements of the connection point and that personalized pair enables implementation of the particular call-back protocol.

7 Claims, 4 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 2

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

□ 5. Document ID: US 6615166 B1

L6: Entry 5 of 63 File: USPT Sep 2, 2003

US-PAT-NO: 6615166
DOCUMENT-IDENTIFIER: US 6615166 B1

TITLE: Prioritizing components of a network framework required for implementation of technology

DATE-ISSUED: September 2, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Guheen; Michael F.	Tiburon	CA		
Mitchell; James D.	Manhattan Beach	CA		
Barrese; James J.	San Jose	CA		

US-CL-CURRENT: 703/27; 703/26, 709/220, 709/223, 709/231, 717/140, 719/316

ABSTRACT:

A system and method are provided for prioritizing components of an existing network framework. First, a plurality of components required for implementation of a predetermined technology using an existing network framework are provided. Next, a priority listing of the components is complied such that the relative position of the components on the priority listing corresponds to a temporal priority among the components. The existing network framework and the components are pictorially represented. Next, a first component of the existing network framework is indicia coded in order to indicate that the first component must be installed first based on the component's position on the priority listing. Thereafter, a second component and any remaining components of the existing network framework is indicia encoded in order to indicate that the second component and any remaining components must be installed after the first component based on the second component's position on the priority listing.

18 Claims, 177 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 177

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Draw. De

---

☐ 6. Document ID: US 6606744 B1

L6: Entry 6 of 63 File: USPT Aug 12, 2003

US-PAT-NO: 6606744
DOCUMENT-IDENTIFIER: US 6606744 B1

TITLE: Providing collaborative installation management in a network-based supply chain environment

DATE-ISSUED: August 12, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY

Mikurak; Michael G.   Hamilton   NJ

US-CL-CURRENT: 717/174; 705/26, 717/178

ABSTRACT:

A system, method and article of manufacture are provided for collaborative installation management in a network-based supply chain environment. According to an embodiment of the invention, telephone calls, data and other multimedia information are routed through a network system which includes transfer of information across the internet utilizing telephony routing information and internet protocol address information. The system includes integrated Internet Protocol (IP) telephony services allowing a user of a web application to communicate in an audio fashion in-band without having to pick up another telephone. Users can click a button and go to a call center through the network using IP telephony. The system invokes an IP telephony session simultaneously with the data session, and uses an active directory lookup whenever a user uses the system. Users include service providers and manufacturers utilizing the network-based supply chain environment.

18 Claims, 130 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 130

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

□ 7. Document ID: US 6601234 B1

L6: Entry 7 of 63   File: USPT   Jul 29, 2003

US-PAT-NO: 6601234
DOCUMENT-IDENTIFIER: US 6601234 B1

TITLE: Attribute dictionary in a business logic services environment

DATE-ISSUED: July 29, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bowman-Amuah; Michel K.	Colorado Springs	CO		

US-CL-CURRENT: 717/108; 705/7, 717/107, 717/116

ABSTRACT:

A system and method are provided for controlling access to data of a business object via an attribute dictionary. The attribute dictionary, which stores attribute names and values, is dispatched over a network. A helper facade is provided for interfacing a business object and the attribute dictionary. Next, it is verified that a current user is authorized to either set or get one of the attribute values upon a request which includes the attribute name that corresponds to the attribute value. The helper facade is called to set, get, or update one of the attribute values based on the corresponding attribute name, wherein the helper facade shields the attribute dictionary from the application code of the business object. The attribute value in the attribute dictionary is obtained or updated if the verification is successful, and a dirty flag is set in the attribute dictionary and an indicator is broadcast upon the attribute value being updated.

15 Claims, 195 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 123

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

□ 8. Document ID: US 6601233 B1

L6: Entry 8 of 63 File: USPT Jul 29, 2003

US-PAT-NO: 6601233
DOCUMENT-IDENTIFIER: US 6601233 B1

TITLE: Business components framework

DATE-ISSUED: July 29, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Underwood; Roy Aaron	Long Grove	IL		

US-CL-CURRENT: 717/102; 717/100, 717/101, 717/103, 717/104, 717/106, 717/107

ABSTRACT:

A method of generating software based on business components. A plurality of logical business components in a business are first defined with each business component having a plurality of capabilities. Next, functional interrelationships are identified between the logical business components. Code modules are then generated to carry out the capabilities of the logical business components and the functional interrelationships between the logical business components, wherein the code modules represent a transformation of the logical business components to their physical implementation, while ensuring the capabilities that are carried out by each code module are essentially unique to the logical business component associated with the code module. Next, the functional aspects of the code modules and the functional relationships of the code modules are tested. The code modules are then subsequently deployed in an e-commerce environment.

18 Claims, 177 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 111

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

□ 9. Document ID: US 6594671 B1

L6: Entry 9 of 63 File: USPT Jul 15, 2003

US-PAT-NO: 6594671
DOCUMENT-IDENTIFIER: US 6594671 B1

TITLE: Separating privileged functions from non-privileged functions in a server instance

DATE-ISSUED: July 15, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Aman; Jeffrey D.	Poughkeepsie	NY		
Frey; Jeffrey A.	New Paltz	NY		
Little; Rodney A.	Poughkeepsie	NY		
Puchkoff; Gary S.	Poughkeepsie	NY		
Riggs; Nancy P.	New Fairfield	CT		

US-CL-CURRENT: 707/103R; 707/10, 719/319

ABSTRACT:

A server instance includes a first region to perform one or more privileged functions and a second region to perform one or more non-privileged functions. Thus, the privileged functions are separate from the non-privileged functions. The first region includes at least a portion of an object request broker, which is used in communicating with one or more clients coupled to the server instance. The second region executes non-privileged application code, and can be replicated within the server instance based on workload management criteria.

40 Claims, 35 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 29

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

□ 10. Document ID: US 6571282 B1

L6: Entry 10 of 63 File: USPT May 27, 2003

US-PAT-NO: 6571282
DOCUMENT-IDENTIFIER: US 6571282 B1

TITLE: Block-based communication in a communication services patterns environment

DATE-ISSUED: May 27, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bowman-Amuah; Michel K.	Colorado Springs	CO		

US-CL-CURRENT: 709/219; 707/10, 709/203, 719/329

ABSTRACT:

A system and method are provided for transmitting data from a server to a client via blocks. A user query for a solution set is transmitted to a server, which is then used to query a database for a first subset of the solution set. A first block of data is built from data in the database of the server, containing a subset of the solution set, and is then sent to the client over a network. When a second request from the client for the data in the database of the server is received, a second block of the data sets is queried from the database, built, and then transmitted to the client over the network.

18 Claims, 195 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 123

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

□ 11. Document ID: US 6557046 B1

L6: Entry 11 of 63 File: USPT Apr 29, 2003

US-PAT-NO: 6557046
DOCUMENT-IDENTIFIER: US 6557046 B1
**** See image for Certificate of Correction ****

TITLE: Method and system for providing an event system infrastructure

DATE-ISSUED: April 29, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
McCauley, III; David E.	Redmond	WA		
Jung; Edward K.	Bellevue	WA		
Heidorn; Christopher S.	Bellevue	WA		

US-CL-CURRENT: 719/318

ABSTRACT:

An event architecture facilitates the asynchronous invocation of events at an event sink. The events are triggered by direct invocation of methods on the event sink. A backchannel connection may be utilized to pass control information from an event sink to an event source. The use of the backchannel connection creates bidirectional connections between an event source and an event sink. A guardian object may be situated between the event source and the event sink to facilitate the asynchronicity of event delivery and to account for disparate threading models between the event source and the event sink. A third party connector may be utilized to connect the event source with the event sink.

19 Claims, 11 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 11

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

□ 12. Document ID: US 6550057 B1

L6: Entry 12 of 63 File: USPT Apr 15, 2003

US-PAT-NO: 6550057
DOCUMENT-IDENTIFIER: US 6550057 B1

TITLE: Piecemeal retrieval in an information services patterns environment

DATE-ISSUED: April 15, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bowman-Amuah; Michel K.	Colorado Springs	CO		

US-CL-CURRENT: 717/126; 700/80, 707/5, 717/101, 717/102, 717/108, 717/109, 717/113

ABSTRACT:

A system, method and article of manufacture are provided for providing a warning upon retrieval of objects that are incomplete. An object is provided with at least one missing attribute. Upon receipt of a request from an application for the object access to the attributes of the object is allowed by the application. A warning is provided upon an attempt to access the attribute of the object that is missing.

15 Claims, 195 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 123

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

□ 13. Document ID: US 6536037 B1

L6: Entry 13 of 63 File: USPT Mar 18, 2003

US-PAT-NO: 6536037
DOCUMENT-IDENTIFIER: US 6536037 B1
**** See image for Certificate of Correction ****

TITLE: Identification of redundancies and omissions among components of a web based architecture

DATE-ISSUED: March 18, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Guheen; Michael F	Tiburon	CA		
Mitchell; James D.	Manhattan Beach	CA		

Barrese; James J.	San Jose	CA

US-CL-CURRENT: 717/151; 703/2, 709/231

ABSTRACT:

A system, method and article of manufacture are provided for conveying redundancies and omissions among components of a network framework such as a web architecture framework. First, an area of an existing network framework is determined in which redundancies and omissions exist. Next, a pictorial representation of the existing network framework is presented along with a plurality of its components. The foregoing redundancies and the omissions are then highlighted by indicia coding the components of the existing network that reside in the area. As such, a diagnostic analysis of redundant efforts and gaps in a current implementation of the existing network framework is effectively conveyed.

19 Claims, 177 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 177

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

□ 14. Document ID: US 6535930 B2

L6: Entry 14 of 63 File: USPT Mar 18, 2003

US-PAT-NO: 6535930
DOCUMENT-IDENTIFIER: US 6535930 B2

TITLE: Method and apparatus for improved interaction with an application program according to data types and actions performed by the application program

DATE-ISSUED: March 18, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Stern; Mark Ludwig	Cupertino	CA		
Johnston, Jr.; Robert George	Cupertion	CA		
Moller; Elizabeth Ann Robinson	Boulder Creek	CA		

US-CL-CURRENT: 719/329

ABSTRACT:

A method and apparatus for performing actions while selecting objects on a user interface display. A user may select a first item in an area of a display controlled by a first process and drag that item to a second area on a display controlled by a second process. The second process may negotiate with the first process to provide certain data types, and based upon those data types, the second process will determine a list of actions in hierarchical fashion which may be performed upon the data. Such actions may include, but are not limited to, printing, sending electronic mail, and performing other actions in the computer system. Alternative and primary actions may also be specified by a user, such as by

selecting various keys on a keyboard or other input device. Further, a user may specify that the second process display a list of actions which may be performed upon the selected item and allow the user to select an action from a list.

8 Claims, 80 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 58

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

□ 15. Document ID: US 6526443 B1

L6: Entry 15 of 63 File: USPT Feb 25, 2003

US-PAT-NO: 6526443
DOCUMENT-IDENTIFIER: US 6526443 B1

TITLE: Method and apparatus for managing transactions with connected computers

DATE-ISSUED: February 25, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Goldsmith; Steven Y.	Albuquerque	NM		
Phillips; Laurence R.	Corrales	NM		
Spires; Shannon V.	Albuquerque	NM		

US-CL-CURRENT: 709/224; 709/218, 719/313

ABSTRACT:

The present invention provides a method and apparatus that make use of existing computer and communication resources and that reduce the errors and delays common to complex transactions such as international shipping. The present invention comprises an agent-based collaborative work environment that assists geographically distributed commercial and government users in the management of complex transactions such as the transshipment of goods across the U.S.-Mexico border. Software agents can mediate the creation, validation and secure sharing of shipment information and regulatory documentation over the Internet, using the World-Wide Web to interface with human users.

25 Claims, 35 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 33

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

□ 16. Document ID: US 6519764 B1

L6: Entry 16 of 63 File: USPT Feb 11, 2003

US-PAT-NO: 6519764
DOCUMENT-IDENTIFIER: US 6519764 B1

TITLE: Method and system for naming and binding objects

DATE-ISSUED: February 11, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Atkinson; Robert G.	Woodinville	WA		
Williams; Antony S.	Mercer Island	WA		
Jung; Edward K.	Seattle	WA		

US-CL-CURRENT: 717/120; 709/217, 719/332

ABSTRACT:

A method and system for referring to and binding to objects using a moniker object is provided. In a preferred embodiments moniker object contains information to identify linked source data and provides methods through which a program can bind to the linked source data. A binding method is provided that returns an instance of an interface through which the linked source data can be accessed. The moniker object can identify source data that is stored persistently or nonpersistently. In addition, moniker objects can be composed to form a composite moniker object. A composite moniker object is used to identify linked source data that is nested in other data. In a preferred embodiment, the moniker object provides other methods including a reducing method that returns a more efficient representation of the moniker object; equality and hash methods for comparing moniker objects; and inverse, common prefix, and relative-path-to methods for comparing and locating moniker objects from other moniker objects. Several implementations of a moniker object are provided including a file moniker, an item moniker, a generic composite moniker, a pointer moniker, and an anti moniker. Each implementation is a moniker class and has a class identifier that identifies code to manage the moniker class.

33 Claims, 68 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 53

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

□ 17. Document ID: US 6507875 B1

L6: Entry 17 of 63 File: USPT Jan 14, 2003

US-PAT-NO: 6507875
DOCUMENT-IDENTIFIER: US 6507875 B1
**** See image for Certificate of Correction ****

TITLE: Modular application collaboration including filtering at the source and proxy execution of compensating transactions to conserve server resources

DATE-ISSUED: January 14, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Mellen-Garnett; Katrina A.	Hillsborough	CA		
Gupta; Prashant	Monterey	CA		

US-CL-CURRENT: 719/310; 718/101

ABSTRACT:

In general, in one aspect, the invention provides a modular application collaborator for providing inter-operability between applications including a plurality of connectors for communicating with a like plurality of applications and an interchange server. The interchange server includes an application collaboration module and service module. The service module transfers messages between connectors and the application collaboration module. The application collaboration defines the inter-operability between two or more applications. The interchange server service module includes a transaction service and an error service. Transactions are executed in the application collaboration module and the transaction service records each action and a compensating action for undoing an associated action. An error service monitors for errors in the interchange server, and, upon detection of an error, stops the execution of a transaction and initiates the execution of any required compensating actions to undo the interrupted transaction. The compensating transactions may be executed at the connectors and are not required to be overseen by the interchange server.

2 Claims, 18 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 15

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

□ 18. Document ID: US 6505210 B1

L6: Entry 18 of 63 File: USPT Jan 7, 2003

US-PAT-NO: 6505210
DOCUMENT-IDENTIFIER: US 6505210 B1

TITLE: Federation of naming contexts across multiple and/or diverse underlying directory technologies

DATE-ISSUED: January 7, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Frey; Jeffrey A.	New Paltz	NY		
Hahn; Timothy J.	Vestal	NY		
Maeurer; Theodore R.	Poughkeepsie	NY		
Puchkoff; Gary S.	Poughkeepsie	NY		

US-CL-CURRENT: 707/103R; 719/310, 719/320

ABSTRACT:

Resolution of a compound name of an object may indicate that a disjunction exists within the object name. This disjunction represents a foreign binding indicating that the resolve cannot be completed on one system or an alias name of the object. When a disjunction in the object name is identified, the object associated with the disjunction is obtained. Thereafter, a resolve operation on that object is performed using at least a portion of the object name.

33 Claims, 35 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 29

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☐ 19. Document ID: US 6502237 B1

L6: Entry 19 of 63 File: USPT Dec 31, 2002

US-PAT-NO: 6502237
DOCUMENT-IDENTIFIER: US 6502237 B1
** See image for Certificate of Correction **

TITLE: Method and apparatus for performing binary translation method and apparatus for performing binary translation

DATE-ISSUED: December 31, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Yates; John S.	Needham	MA		
Tye; Steven Tony	Hopkinton	MA		
Hookway; Raymond J.	West Boyslston	MA		

US-CL-CURRENT: 717/136

ABSTRACT:

A computer system for executing a binary image conversion system which converts instructions from a instruction set of a first, non native computer system to a second, different, native computer system, includes an run-time system which in response to a non-native image of an application program written for a non-native instruction set provides an native instruction or a native instruction routine. The run-time system collects profile data in response to execution of the native instructions to determine execution characteristics of the non-native instruction. Thereafter, the non-native instructions and the profile statistics are fed to a binary translator operating in a background mode and which is responsive to the profile data generated by the run-time system to form a translated native image. The run-time system and the binary translator are under the control of a server process. The non-native image is executed in two different environments with first portion executed as an interpreted image and remaining portions as a translated image. The run-time system includes an interpreter which is capable of handling condition codes corresponding to the non-native architecture. A technique is also provided to jacket calls between the two execution environments and to support object based services. Preferred techniques are also provided to determine interprocedural translation units. Further, intermixed translation/optimization techniques are discussed.

9 Claims, 87 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 72

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

□ 20. Document ID: US 6502103 B1

L6: Entry 20 of 63 File: USPT Dec 31, 2002

US-PAT-NO: 6502103
DOCUMENT-IDENTIFIER: US 6502103 B1

TITLE: Providing composed containers and data objects to support multiple resources

DATE-ISSUED: December 31, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Frey; Jeffrey A.	New Paltz	NY		
Fulkerson, Jr.; Carroll E.	Raleigh	NC		
Little; Rodney A.	Poughkeepsie	NY		
Puchkoff; Gary S.	Poughkeepsie	NY		

US-CL-CURRENT: 707/103R; 718/100, 719/310

ABSTRACT:

A composed business object of a server instance is provided. The server instance is coupled to a plurality of resource managers using a container of the server instance. A data object corresponding to the business object is populated with data retrieved using the plurality of resource managers. Thus, a business object is composed of data from multiple and/or diverse resource managers using a single container.

30 Claims, 35 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 29

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

□ 21. Document ID: US 6457065 B1

L6: Entry 21 of 63 File: USPT Sep 24, 2002

US-PAT-NO: 6457065

DOCUMENT-IDENTIFIER: US 6457065 B1

TITLE: Transaction-scoped replication for distributed object systems

DATE-ISSUED: September 24, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Rich; Lawrence Scott	Apex	NC		
Salo; Timo J.	Apex	NC		

US-CL-CURRENT: 719/328; 718/101, 719/329, 719/330

ABSTRACT:

A method, system, and computer program product for improving the performance of distributed object systems. A remote object is replicated to the node of the distributed system from which it is accessed. The scope of the replication is a transaction. Thereafter, method invocations on the object occur locally, avoiding the performance overhead of frequent round trips to the remote persistent object store. Changes made to a replicated object by a transaction are represented using a tree structure that is internally managed by the application. When an application or application user has made modifications to a replicated object and requests to commit the modifications, a determination is first made as to whether committing the modifications will result in an unacceptable data conflict. If no unacceptable data conflict will occur, and after resolution of those conflicts that can be resolved, the modifications are committed. Nested transactions are supported, where each child transaction may commit or roll back independently.

24 Claims, 14 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 14

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

□ 22. Document ID: US 6446256 B1

L6: Entry 22 of 63 File: USPT Sep 3, 2002

US-PAT-NO: 6446256
DOCUMENT-IDENTIFIER: US 6446256 B1

TITLE: Extension of parsable structures

DATE-ISSUED: September 3, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Hyman; Michael I.	Seattle	WA		
Vaddadi; Phani K.	Issaquah	WA		

US-CL-CURRENT: 717/143; 715/513, 715/515, 715/516, 715/524, 717/115, 717/116, 717/140, 717/141, 717/142, 717/147, 717/165

ABSTRACT:

A bilateral interface may comprise an object interface and, optionally, a language interface. The object interface comprises at least an object-side object interface to be included in an external object, and a structure-side object interface to be included in a parsable structure. When executed, an object initialization routine included in the object-side object interface provides an identification of the structure-side object interface to the external object. Conversely, a structure initialization routine included in the structure-side object interface provides an identification of at least one object interaction routine to the external object. The external object can then invoke the at least one object interaction routine to interact with the parsable structure. The language interface is similarly constructed. The bilateral interface of the present invention provides much greater extensibility for parsable structures than was previously possible and provides, for example, for the addition of customized keywords.

11 Claims, 9 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 7

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

□ 23. Document ID: US 6442748 B1

L6: Entry 23 of 63 File: USPT Aug 27, 2002

US-PAT-NO: 6442748
DOCUMENT-IDENTIFIER: US 6442748 B1

TITLE: System, method and article of manufacture for a persistent state and persistent object separator in an information services patterns environment

DATE-ISSUED: August 27, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bowman-Amuah; Michel K.	Colorado Springs	CO		

US-CL-CURRENT: 717/108; 707/103R, 707/104.1, 707/9, 719/316

ABSTRACT:

A system, method and article of manufacture are provided for separating logic and data access concerns during development of a persistent object for insulating development of business logic from development of data access routine. A persistent object being developed is accessed and a state of the persistent object is detached into a separate state class. The state class serves as a contract between a logic development team and a data access development team. Logic development is limited by the logic development team to developing business logic. Data access development is restricted by the data access development team to providing data creation, retrieval, updating, and deletion capabilities.

18 Claims, 195 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 123

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

□ 24. Document ID: US 6437805 B1

L6: Entry 24 of 63 File: USPT Aug 20, 2002

US-PAT-NO: 6437805
DOCUMENT-IDENTIFIER: US 6437805 B1

TITLE: System and method for accessing object capabilities in a graphical program

DATE-ISSUED: August 20, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Sojoodi; Omid	Austin	TX		
Dye; Robert	Austin	TX		
Parthasarathy; Murali	Austin	TX		
Kudukoli; Ram	Austin	TX		

US-CL-CURRENT: 345/763; 717/105

ABSTRACT:

A system and method for creating a graphical program, wherein the graphical program is operable to access capabilities of an object. During creation of the graphical program, the user operates to place an object node in the graphical program, wherein the object node is operable to access capabilities of the object. This preferably includes the user arranging on the screen the graphical program, including the object node and various other nodes, and connecting the various nodes to create the graphical program. The user then configures the object node to receive information on the object, preferably by the user configuring the object node with a reference to the object, e.g., a pointer, address, or other information which specifies the identity and/or location of the object. The user also selects one or more methods to be invoked on the object and/or one or more properties to get/set on the object. Once the graphical program has been created, then during execution of the graphical program, the object node accesses the capabilities of the object.

84 Claims, 63 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 40

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw De

□ 25. Document ID: US 6425017 B1

L6: Entry 25 of 63 File: USPT Jul 23, 2002

US-PAT-NO: 6425017
DOCUMENT-IDENTIFIER: US 6425017 B1

TITLE: Queued method invocations on distributed component applications

DATE-ISSUED: July 23, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Dievendorff; Richard	Bellevue	WA		
Helland; Patrick J.	Bellevue	WA		
Chopra; Gagan	Redmond	WA		
Al-Ghosein; Mohsen	Redmond	WA		

US-CL-CURRENT: 719/315

ABSTRACT:

An object runtime architecture allows method invocations to be made on either a synchronous, real-time basis or a queued basis using the normal call semantics of an object model. The object runtime architecture provides a proxy of an object with a method invocation recorder for receiving method calls of a client on the object, and marshaling the method calls into a message for sending to a queue associated with the object. The object runtime architecture further provides a listener for dispatching the message from the queue to a player which uses a stub to unmarshal the message in order to issue the method calls to the object. The object runtime architecture thus decouples the client and objects lifetimes and availability, without requiring explicit programming of the client and object to perform message queuing. Accordingly, with no modification of the object's interface structure or code, the same object can be used in either a real-time or queued environment. This allows the decision between real-time or queued method invocations to be made much later than at development of the object, such as at run-time creation of the object.

14 Claims, 10 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 7

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

□ 26. Document ID: US 6424991 B1

L6: Entry 26 of 63 File: USPT Jul 23, 2002

US-PAT-NO: 6424991
DOCUMENT-IDENTIFIER: US 6424991 B1

TITLE: Object-oriented system, method and article of manufacture for a client-server communication framework

DATE-ISSUED: July 23, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Gish; Sheri L.	Mountain View	CA		

US-CL-CURRENT: 709/203; 709/227, 719/310

ABSTRACT:

An interprise computing manager in which an application is composed of a client (front end) program which communicates utilizing a network with a server (back end) program. The client and server programs are loosely coupled and exchange information using the network. The client program is composed of a User Interface (UI) and an object-oriented framework (Presentation Engine (PE) framework). The UI exchanges data messages with the framework. The framework is designed to handle two types of messages: (1) from the UI, and (2) from the server (back end) program via the network. The framework includes a component, the mediator which manages messages coming into and going out of the framework. The system includes software for a client computer, a server computer and a network for connecting the client computer to the server computer which utilize an execution framework code segment configured to couple the server computer and the client computer via the network, by a plurality of client computer code segments resident on the server, each for transmission over the network to a client computer to initiate coupling; and a plurality of server computer code segments resident on the server which execute on the server in response to initiation of coupling via the network with a particular client utilizing the transmitted client computer code segment for communicating via a particular communication protocol. A session manager is utilized to manage the initiation of a plurality of network sessions.

19 Claims, 30 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 17

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☐ 27. Document ID: US 6415434 B1

L6: Entry 27 of 63 File: USPT Jul 2, 2002

US-PAT-NO: 6415434
DOCUMENT-IDENTIFIER: US 6415434 B1

TITLE: Apparatus and method for a runtime method overloading resolver

DATE-ISSUED: July 2, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Kind; Lance	Fort Collins	CO		

US-CL-CURRENT: 717/107; 717/106, 717/108, 717/114, 717/116, 717/131, 717/133, 717/162, 717/165, 717/166

ABSTRACT:

A method and apparatus for resolving method overloading at runtime that includes accessing an application programming interface file to retrieve methods that belong to the same class as a target method to select an exact method, which is a method where a data type of each of the parameters of the exact method is the same data type of a corresponding parameter of the target method, and if there is no exact method, to find a best method, which is a best method that most closely matches the target method.

20 Claims, 5 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 5

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

□ 28. Document ID: US 6412010 B1

L6: Entry 28 of 63 File: USPT Jun 25, 2002

US-PAT-NO: 6412010
DOCUMENT-IDENTIFIER: US 6412010 B1
**** See image for Certificate of Correction ****

TITLE: APPARATUS AND METHOD FOR IMPLEMENTING A NETWORK PROTOCOL THAT SUPPORTS THE TRANSMISSION OF A VARIABLE NUMBER OF APPLICATION-USABLE OBJECT OVER A NETWORK AS A SINGLE NETWORK TRANSMITTABLE CONTAINER OBJECT AND THE RE-CREATION OF THOSE APPLICATION-USABLE OBJECT THEREFROM

DATE-ISSUED: June 25, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Kind; Lance	Fort Collins	CO		

US-CL-CURRENT: 709/230; 717/108, 717/116

ABSTRACT:

A method and apparatus for implementing a network protocol to support the transmission of a variable number of application-usable objects as a single network transmittable container object and the re-creation of the application-usable objects therefrom, which includes marshalling application-usable objects into marshalled objects, where each application-usable objects corresponds to a different marshalled object, and each marshalled object comprises frames of data; creating a two-dimensional array to transmit the application-usable objects, where a first dimension represents a given marshalled object by referencing a second dimension array, and a second dimension array comprises frames of data of a given marshalled object; and storing each marshalled object in the two-dimensional array.

25 Claims, 3 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 3

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

☐ 29. Document ID: US 6405364 B1

L6: Entry 29 of 63 File: USPT Jun 11, 2002

US-PAT-NO: 6405364
DOCUMENT-IDENTIFIER: US 6405364 B1

TITLE: Building techniques in a development architecture framework

DATE-ISSUED: June 11, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bowman-Amuah; Michel K.	Colorado Springs	CO		

US-CL-CURRENT: 717/101; 717/102, 717/120, 717/124

ABSTRACT:

A system is provided for building systems in a development architecture framework. The present invention is directed to both a system to be built and an implementation strategy to fulfill system requirements. Software components of the system are encapsulated with wrappers. The wrappers are adapted to be changed upon other software components of the system being changed while the encapsulated software components of the system remain unchanged. In one embodiment of the present invention, specifying the requirements of the system to be built and the implementation strategy to fulfill the requirements may be carried out using tools such as data modeling tools, process modeling tools, event modeling tools, performance modeling tools, object modeling tools, component modeling tools, reuse support tools, prototyping tools, application logic design tools, database design tools, presentation design tools, communication design, and usability test tools. In another embodiment of the present invention, improving the performance and maintenance of the system may be carried out using tools such as interactive navigation tools, graphical representation tools, extraction tools, repository tools, restructuring tools, and data name rationalization tools.

12 Claims, 14 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 14

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

☐ 30. Document ID: US 6356933 B2

L6: Entry 30 of 63 File: USPT Mar 12, 2002

US-PAT-NO: 6356933
DOCUMENT-IDENTIFIER: US 6356933 B2

TITLE: Methods and apparatus for efficiently transmitting interactive application data between a client and a server using markup language

DATE-ISSUED: March 12, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Mitchell; David C.	Orem	UT		
Walters; Ben	Salt Lake City	UT		
Mitchell; Dale	Orem	UT		

US-CL-CURRENT: 709/203; 345/762, 715/513, 719/315, 719/328, 719/329

ABSTRACT:

A method for efficiently transferring data between a client and a server includes the steps of: providing an application program; providing an application-independent client process effecting a plurality of client states; providing an application-independent server process effecting a plurality of server states; transferring data from the server process to the client process in response to an application program; and updating at least one client state in response to the transferred data. A related apparatus is also disclosed.

20 Claims, 6 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 6

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

□ 31. Document ID: US 6353923 B1

L6: Entry 31 of 63 File: USPT Mar 5, 2002

US-PAT-NO: 6353923
DOCUMENT-IDENTIFIER: US 6353923 B1

TITLE: Active debugging environment for debugging mixed-language scripting code

DATE-ISSUED: March 5, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bogle; Phillip Lee	Redmond	WA		
Katzenberger; Gary S.	Woodinville	WA		
McKelvie; Samuel James	Seattle	WA		
Welland; Robert Victor	Seattle	WA		

US-CL-CURRENT: 717/128

ABSTRACT:

An active debugging environment for debugging a virtual application that contains program language code from multiple compiled and/or interpreted programming languages. The active debugging environment is language neutral and host neutral, where the host is a standard content centric script host with language engines for each of the multiple compiled and/or interpreted programming languages represented in the virtual application. The active debugging environment user interface can be of any debug tool interface design. The language neutral and host neutral active debugging environment is facilitated by a process debug manager that catalogs and manages application specific components, and a machine debug manager that catalogs and manages the various applications that comprise a virtual application being run by the script host. The process debug manager and the machine debug manager act as an interface between the language engine specific programming language details and the debug user interface.

20 Claims, 7 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 6

□ 32. Document ID: US 6332163 B1

L6: Entry 32 of 63 File: USPT Dec 18, 2001

US-PAT-NO: 6332163
DOCUMENT-IDENTIFIER: US 6332163 B1

TITLE: Method for providing communication services over a computer network system

DATE-ISSUED: December 18, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bowman-Amuah; Michel K.	Colorado Springs	CO		

US-CL-CURRENT: 709/231; 709/217, 709/223, 709/227, 719/329

ABSTRACT:

A system, method and article of manufacture are provided for implementing communication services patterns. A fixed format stream-based communication system is provided and service is delivered via a globally addressable interface. Access is afforded to a legacy system. Service is delivered via a locally addressable interface. A null value is communicated and data is transmitted from a server to a client via pages. A naming service and a client are interfaced with the naming service allowing access to a plurality of different sets of services from a plurality of globally addressable interfaces. A stream-based communication system is provided and data is efficiently retrieved.

15 Claims, 195 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 123

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

□ 33. Document ID: US 6272537 B1

L6: Entry 33 of 63 File: USPT Aug 7, 2001

US-PAT-NO: 6272537
DOCUMENT-IDENTIFIER: US 6272537 B1

TITLE: Method for building element manager for a computer network element using a visual element manager builder process

DATE-ISSUED: August 7, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Kekic; Miodrag M.	Mountain View	CA		
Lu; Grace N.	Milpitas	CA		
Carlton; Eloise H.	San Carlos	CA		

US-CL-CURRENT: 709/223; 709/203, 709/219, 719/329

ABSTRACT:

A client-server network management system includes: a plurality of managed computer network elements, a managed element server that executes on a first computer; and at least one managed element server client that typically executes on a second computer. The managed element server and managed element server client are computer processes that execute from memory of their respective computers. The client-server network management system is really two applications in one: a visual element manager builder and a manager. The manager provides the run-time environment in which element managers are executed to monitor and manage computer network behavior such as network throughput, collision rate, and number of duplicate IP packets, to name a few. The manager portion of managed element server is independent of any graphic user interface. The logic and structure of the manager of managed element server is cleanly separated from the graphic user interfaces. The visual element manager builder is a visual development environment in which device vendors or network managers may create standardized element management applications, called element managers. A user can build an element manager without writing a any computer code. In addition, a user can edit an element manager without writing any computer code. A graphic user interface of this invention, that is displayed by the client, includes a visual image of a computer network element being managed. As a user looks at the visual display in the graphic user interface, the user is provided the same visual information as if the user where physically present at the location of the managed computer network element. Thus, at a glance, a user can obtain considerable information about the status of the computer network element as represented by the visual display.

26 Claims, 73 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 50

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☐ 34. Document ID: US 6269373 B1

L6: Entry 34 of 63 File: USPT Jul 31, 2001

US-PAT-NO: 6269373
DOCUMENT-IDENTIFIER: US 6269373 B1

TITLE: Method and system for persisting beans as container-managed fields

DATE-ISSUED: July 31, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Apte; Ajay Arvind	Austin	TX		
Acker; Liane Elizabeth	Round Rock	TX		

US-CL-CURRENT: 707/10; 719/311, 719/316, 719/326

ABSTRACT:

A method for persisting a container-managed server object or bean in a distributed data processing system is provided. A method in a Tie object is invoked by the container of the bean to assist in the persistence of the bean. The bean may have container-managed fields of both primitive and complex data-types, and the complex data-types may include references to other objects or beans. The Tie object obtains the object reference to the other object. The Tie object unwraps an adapter from the object reference to obtain a proxy of the referred object and stringifies the proxy. The string is then stored by the container in a back-end data store as a known primitive data-type. The referred object may be resurrected by unstringifying the string and reversing the process.

17 Claims, 19 Drawing figures
Exemplary Claim Number: 9
Number of Drawing Sheets: 8

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

☐ 35. Document ID: US 6266726 B1

L6: Entry 35 of 63 File: USPT Jul 24, 2001

US-PAT-NO: 6266726
DOCUMENT-IDENTIFIER: US 6266726 B1

TITLE: Process control system using standard protocol control-of standard devices and non-standard devices

DATE-ISSUED: July 24, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY

Nixon; Mark	Round Rock	TX		
Havekost; Robert B.	Austin	TX		
Jundt; Larry O.	Round Rock	TX		
Stevenson; Dennis	Round Rock	TX		
Ott; Michael G.	Austin	TX		
Webb; Arthur	Leicestershire			GB
Lucas; Mike	Leicestershire			GB

US-CL-CURRENT: 710/105; 713/1, 713/100, 719/313

ABSTRACT:

A process controller implements and executes a standard set of function blocks or control functions defined by a standard protocol so that standard-type control is achieved with respect to non-standard-type devices. The process controller enables standard devices to implement the standard set of function blocks and control functions. The process controller implements an overall strategy as if all connected devices are standard devices by usage of a Fieldbus function block as a fundamental building block for control structures. These function blocks are defined to create control structures for all types of devices.

10 Claims, 25 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 25

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

□ 36. Document ID: US 6266716 B1

L6: Entry 36 of 63 File: USPT Jul 24, 2001

US-PAT-NO: 6266716
DOCUMENT-IDENTIFIER: US 6266716 B1

TITLE: Method and system for controlling data acquisition over an information bus

DATE-ISSUED: July 24, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Wilson; Douglass J.	Boston	MA		
Colan; Mark T.	Medford	MA		

US-CL-CURRENT: 710/33; 709/213, 709/216, 719/312, 719/313

ABSTRACT:

A group of protocols is described that establish an information bus. The protocols allow various applications and components to plug into the information bus. As a member of the bus, each application or component can exchange information with any other application or component in a structured way. The information bus is especially useful in interconnecting Java beans and applets in a Java virtual machine and in a distributive computer environment. An apparatus is disclosed that is utilized to produce data from an application to an information bus for sharing the data with other applications utilizing the information bus. The data producing apparatus comprises a data element building logic, which builds data elements containing the data within the application, a data notification logic, which notifies the information bus of the availability of the data element, and a data element transfer logic, which transfers data element from one application to another. An event listing logic can also be included that is utilized to listen for data element requests from applications utilizing the information bus. The data element builder logic may be configured to change the data within a data element and the data notification logic may be configured to announce the data change across the information bus to at least some of the applications. The data element builder logic specifies the data name using either a property or a parameter. The data element builder logic also provides a view associated with the data item where the data item is a java object. The data element builder logic may also remove data elements for access while the data notification logic notifies the information bus of the removal of the particular data elements.

24 Claims, 14 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 7

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

□ 37. Document ID: US 6263379 B1

L6: Entry 37 of 63 File: USPT Jul 17, 2001

US-PAT-NO: 6263379
DOCUMENT-IDENTIFIER: US 6263379 B1

TITLE: Method and system for referring to and binding to objects using identifier objects

DATE-ISSUED: July 17, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Atkinson; Robert G.	Woodinville	WA		
Williams; Antony S.	Mercer Island	WA		
Jung; Edward K.	Seattle	WA		

US-CL-CURRENT: 719/332

ABSTRACT:

A method and system for referring to and binding to objects using a moniker object is provided. In a preferred embodiments moniker object contains information to identify linked source data and provides methods through which a program can bind to the linked source data. A binding method is provided that returns an instance of an interface through which the linked source data can be accessed. The moniker object can identify source data that is stored persistently or nonpersistently. In addition, moniker objects can be composed to form a composite moniker object. A composite moniker object is used to identify linked source data that is nested in

other data. In a preferred embodiment, the moniker object provides other methods including a reducing method that returns a more efficient representation of the moniker object; equality and hash methods for comparing moniker objects; and inverse, common prefix, and relative-path-to methods for comparing and locating moniker objects from other moniker objects. Several implementations of a moniker object are provided including a file moniker, an item moniker, a generic composite moniker, a pointer moniker, and an anti moniker. Each implementation is a moniker class and has a class identifier that identifies code to manage the moniker class.

17 Claims, 68 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 53

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

□ 38. Document ID: US 6260186 B1

L6: Entry 38 of 63 File: USPT Jul 10, 2001

US-PAT-NO: 6260186
DOCUMENT-IDENTIFIER: US 6260186 B1
**** See image for Certificate of Correction ****

TITLE: Universal state machine for use with a concurrent state machine space in a telecommunications network

DATE-ISSUED: July 10, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
James; Daniel	Richardson	TX		

US-CL-CURRENT: 717/104

ABSTRACT:

A state machine space for concurrent state machines is provided for use in a telecommunications system. Universal State Machines (USMs) use a universal data structure (representing an event in the telecommunications system) as a way for the system architect to dynamically alter the how the events are handled. The USMs are designed to function concurrently with other USMs without requiring explicit knowledge of other USMs or their functionality. A USM definition (USMD) defines a state machine and contains configuration, states and transitions information. An instantiation of a USM includes a USM definition pointer, a current state indicator, a current event processor indicator, and optional lists of event processors and spawned child state machines (other USMDs). Other state machines may be hooked or referenced by a main state machine in its definition. Upon instantiation of the main state machine, other state machines may be hooked or referenced and processing control is transferred to the hooked or referenced state machines to perform functions.

22 Claims, 26 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 23

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

□ 39. Document ID: US 6256773 B1

L6: Entry 39 of 63 File: USPT Jul 3, 2001

US-PAT-NO: 6256773
DOCUMENT-IDENTIFIER: US 6256773 B1

TITLE: System, method and article of manufacture for configuration management in a development architecture framework

DATE-ISSUED: July 3, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bowman-Amuah; Michel K.	Colorado Springs	CO		

US-CL-CURRENT: 717/121; 707/203, 717/168

ABSTRACT:

A system, method, and article of manufacture are provided for affording consistency in a development architecture framework as components in the framework change. A reference program code is provided and a plurality of sets of updated program code are received which represent different versions of the program code. The sets of the updated program code are compared with the reference program code in order to identify information relating to changes and the information is classified in relation to the changes. Tools are also provided for managing the different versions of the program code.

20 Claims, 14 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 14

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

□ 40. Document ID: US 6253253 B1

L6: Entry 40 of 63 File: USPT Jun 26, 2001

US-PAT-NO: 6253253
DOCUMENT-IDENTIFIER: US 6253253 B1

TITLE: Method and apparatus for optimizing references to objects in a data processing system

DATE-ISSUED: June 26, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Mason; Carlton Keith	Austin	TX		
Kodeih; Mohamad	Austin	TX		
High, Jr.; Robert Howard	Round Rock	TX		

US-CL-CURRENT: 719/315

ABSTRACT:

A method and apparatus for optimizing references to objects in a distributed data processing system. A method is invoked in a client process by client application on a target object. In response to determining that the target object is on a remote process reference, a smart proxy determines whether the message can be processed within the client process. In response to determining that the message can be processed in the client process, the message is processed locally. Otherwise the message is sent to the target object for processing.

5 Claims, 8 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 8

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

□ 41. Document ID: US 6226789 B1

L6: Entry 41 of 63 File: USPT May 1, 2001

US-PAT-NO: 6226789
DOCUMENT-IDENTIFIER: US 6226789 B1
**** See image for Certificate of Correction ****

TITLE: Method and apparatus for data flow analysis

DATE-ISSUED: May 1, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Tye; Steven Tony	Hopkinton	MA		
Yates; John S.	Needham	MA		

US-CL-CURRENT: 717/138; 717/155, 717/158

ABSTRACT:

A computer system for executing a binary image conversion system which converts instructions from a instruction set of a first, non native computer system to a second, different, native computer system, includes an run-time system which in response to a non-native image of an application program written for a non-native instruction set provides an native instruction or a native instruction routine. The run-time system collects profile data in response to execution of the native instructions to determine execution characteristics of the non-native instruction.

Thereafter, the non-native instructions and the profile statistics are fed to a binary translator operating in a background mode and which is responsive to the profile data generated by the run-time system to form a translated native image. The run-time system and the binary translator are under the control of a server process. The non-native image is executed in two different enviroments with first portion executed as an interpreted image and remaining portions as a translated image. The run-time system includes an interpreter which is capable of handling condition codes corresponding to the non-native architecute. A technique is also provided to jacket calls between the two execution enviroments and to support object based services. Preferred techniques are also provide to determine interprocedural translation units. Further, intermixed translation/optimization techniques are discussed.

15 Claims, 87 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 72

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

□ 42. Document ID: US 6226692 B1

L6: Entry 42 of 63 File: USPT May 1, 2001

US-PAT-NO: 6226692
DOCUMENT-IDENTIFIER: US 6226692 B1

TITLE: Method and system for constructing software components and systems as assemblies of independent parts

DATE-ISSUED: May 1, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Miloushev; Vladimir I.	Laguna Niguel	CA		
Nickolov; Peter A.	Irvine	CA		

US-CL-CURRENT: 719/316

ABSTRACT:

A system and a method for designing and constructing software components and systems by assembling them from independent parts which is compatible with and extends existing object models. A terminal interface and a terminal mechanism for interfacing objects is included. The mechanism is independent from the actual type of interactions established through it and allows objects to invoke directly services of other objects. All objects in a given system implement and expose a terminal interface. A property interface and mechanism with hierarchical property names and ability to execute queries is also included. The mechanism can be used for parameterization and serialization of objects, as well as to provide structured storage. A new and advantageous type of software object, named parts, is defined. Parts are constructed through an abstract factory and implement a property interface and a terminal interface.

7 Claims, 58 Drawing figures

Exemplary Claim Number: 1
Number of Drawing Sheets: 58

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☐ 43. Document ID: US 6226666 B1

L6: Entry 43 of 63 File: USPT May 1, 2001

US-PAT-NO: 6226666
DOCUMENT-IDENTIFIER: US 6226666 B1

TITLE: Agent-based management system having an open layered architecture for synchronous and/or asynchronous messaging handling

DATE-ISSUED: May 1, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Chang; Daniel T.	San Jose	CA		
Sundaresan; Neelakantan	San Jose	CA		

US-CL-CURRENT: 709/202; 379/100.08, 379/93.24, 709/201, 709/206, 719/315

ABSTRACT:

A communication infrastructure providing communication between agents, between agents and agent-hosting servers, and between agent-hosting servers. The communication infrastructure consists of three layers (from bottom to top): Mail Facility Layer, Message Facility Layer, and Agent Management Communication Facility Layer. The Mail Facility Layer is the lowest layer providing a general, semantics-free mail paradigm for asynchronous communication between distributed objects, whether they are local or remote to each other. The Mail Facility Layer provides a level of abstraction in terms of mail, virtual mailbox, post office, and mail queue, and hides the details of implementation and actual transport. It is designed to provide location transparency and to be implementable using various transport protocols. The next Message Facility Layer provides a typed messaging paradigm for asynchronous and synchronous message passing between distributed objects. The Message Facility Layer uses the Mail Facility Layer for sending messages and for getting responses to requests sent. It allows for the association of typed message handlers with typed messages such that the format and semantics of messages are encapsulated through their types, are extensible, and can be processed by the associated message handlers. The Agent Management Communication Facility Layer is the highest layer providing the services for inter-agent communication between agents, agent-agent-server communication between an agent and an agent server, and inter-agent-server communication between agent servers for managing agents such as locating an agent, dispatching an agent, retrieving an agent, etc. The key abstractions provided in this layer include agent manager, agent, and agent identifier. It uses the Message

12 Claims, 28 Drawing figures
Exemplary Claim Number: 9
Number of Drawing Sheets: 16

□ 44. Document ID: US 6212577 B1

L6: Entry 44 of 63 File: USPT Apr 3, 2001

US-PAT-NO: 6212577
DOCUMENT-IDENTIFIER: US 6212577 B1

TITLE: Method and apparatus for improved interaction with an application program according to data types and actions performed by the application program

DATE-ISSUED: April 3, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Stern; Mark Ludwig	Cupertino	CA		
Johnston, Jr.; Robert George	Cupertino	CA		
Robinson Moller; Elizabeth Ann	Boulder Creek	CA		

US-CL-CURRENT: 719/329

ABSTRACT:

A method and apparatus for performing actions while selecting objects on a user interface display. A user may select a first item in an area of a display controlled by a first process and drag that item to a second area on a display controlled by a second process. The second process may negotiate with the first process to provide certain data types, and based upon those data types, the second process will determine a list of actions in hierarchical fashion which may be performed upon the data. Such actions may include, but are not limited to, printing, sending electronic mail, and performing other actions in the computer system. Alternative and primary actions may also be specified by a user, such as by selecting various keys on a keyboard or other input device. Further, a user may specify that the second process display a list of actions which may be performed upon the selected item and allow the user to select an action from a list.

15 Claims, 80 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 58

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

□ 45. Document ID: US 6212574 B1

L6: Entry 45 of 63 File: USPT Apr 3, 2001

US-PAT-NO: 6212574
DOCUMENT-IDENTIFIER: US 6212574 B1

TITLE: User mode proxy of kernel mode operations in a computer operating system

DATE-ISSUED: April 3, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
O'Rourke; Thomas J.	Oulu			FI
Shaw; George H. J.	Woodinville	WA		
Woodruff; Bryan A.	New Bend	WA		

US-CL-CURRENT: 719/321

ABSTRACT:

This invention involves user mode proxy of kernel mode operations in a computer operating system. The broad contours of the invention allow drivers operating in the kernel mode of an operating system to be proxied by a corresponding user mode object. A user mode process wishing to communicate with or manipulate the kernel mode driver may perform such communication or manipulation by communicating with or manipulating the corresponding user mode proxy. The present invention has particular applicability to emerging operating systems which allow kernel mode drivers to be interconnected in order to create a processing stream in the kernel where data is passed from kernel mode driver to kernel mode driver without transitioning to user mode. User mode proxy of such kernel mode drivers allows user mode processes to manipulate such kernel mode drivers using familiar user mode protocols without the necessity of learning new kernel mode protocols.

18 Claims, 10 Drawing figures
Exemplary Claim Number: 11
Number of Drawing Sheets: 9

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

□ 46. Document ID: US 6145013 A

L6: Entry 46 of 63 File: USPT Nov 7, 2000

US-PAT-NO: 6145013
DOCUMENT-IDENTIFIER: US 6145013 A

TITLE: Distributed instantiation system and method

DATE-ISSUED: November 7, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Morningstar; Chip	Palo Alto	CA		
Farmer; F. Randall	Palo Alto	CA		

US-CL-CURRENT: 719/316

ABSTRACT:

A communication system and method includes unums distributed over at least a single presence and including a selected plurality of ingredients. An unum is established by creating ingredients at the level of its interface and attributes; and at the level of its implementation; and interconnecting ingredients into presences and unums. Communication between ingredients is accomplished within a single presence, across an unum boundary within an agency, or within a single unum across a presence boundary. Trust boundaries are established between presences and unums to establish a predetermined level of communications security in messaging between ingredients.

8 Claims, 18 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 16

Full | Title | Citation | Front | Review | Classification | Date | Reference | [illegible] | [illegible] | Claims | KWIC | Draw De

□ 47. Document ID: US 6115040 A

L6: Entry 47 of 63 File: USPT Sep 5, 2000

US-PAT-NO: 6115040
DOCUMENT-IDENTIFIER: US 6115040 A

TITLE: Graphical user interface for Web enabled applications

DATE-ISSUED: September 5, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bladow; Chad R.	Monument	CO		
Devine; Carol Y.	Colorado Springs	CO		
Schwarz; Edward	New York	NY		
Shamash; Arieh	Great Neck	NY		
Shoulberg; Richard W.	Manitou Springs	CO		
Wood; Jeffrey A.	Colorado Springs	CO		

US-CL-CURRENT: 345/741; 345/969, 709/203, 709/217, 719/315

ABSTRACT:

An integrated system of user interfaces for communicating with remote services. A backplane architecture controls and manages the user interfaces by instantiating, launching, overseeing and closing the user interfaces associated with a plurality of applications residing in a remote server. Each application communicates with one another and with the backplane via messaging interfaces. The backplane provides a single uniform user authentication procedure during logon for the user interfaces and also provides session management for a duration of a user session. Session information is maintained by exchanging a session key or keys for identifying the session with remote servers.

29 Claims, 12 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 12

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

□ 48. Document ID: US 6094688 A

L6: Entry 48 of 63 File: USPT Jul 25, 2000

US-PAT-NO: 6094688
DOCUMENT-IDENTIFIER: US 6094688 A

TITLE: Modular application collaboration including filtering at the source and proxy execution of compensating transactions to conserve server resources

DATE-ISSUED: July 25, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Mellen-Garnett; Katrina A.	Hillsborough	CA		
Gupta; Prashant	Monterey	CA		

US-CL-CURRENT: 719/328; 718/101, 719/318

ABSTRACT:

In general, in one aspect, the invention provides a modular application collaborator for providing inter-operability between applications including a plurality of connectors for communicating with a like plurality of applications and an interchange server. The interchange server includes an application collaboration module and service module. The service module transfers messages between connectors and the application collaboration module. The application collaboration defines the inter-operability between two or more applications. The interchange server service module includes a transaction service and an error service. Transactions are executed in the application collaboration module and the transaction service records each action and a compensating action for undoing an associated action. An error service monitors for errors in the interchange server, and, upon detection of an error, stops the execution of a transaction and initiates the execution of any required compensating actions to undo the interrupted transaction. The compensating transactions may be executed at the connectors and are not required to be overseen by the interchange server.

16 Claims, 16 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 15

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

□ 49. Document ID: US 6091897 A

L6: Entry 49 of 63 File: USPT Jul 18, 2000

US-PAT-NO: 6091897

DOCUMENT-IDENTIFIER: US 6091897 A
**** See image for Certificate of Correction ****

TITLE: Fast translation and execution of a computer program on a non-native architecture by use of background translator

DATE-ISSUED: July 18, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Yates; John S.	Needham	MA		
Robinson; Scott G.	Tyngsboro	MA		
Herdeg; Mark	Leominster	MA		

US-CL-CURRENT: 717/138; 717/153, 717/158

ABSTRACT:

A computer system for executing a binary image conversion system which converts instructions from a instruction set of a first, non native

computer system to a second, different native computer system, includes an run-time system which in response to a non-native image of an application program written for a non-native instruction set provides an native instruction or a native instruction routine. The run-time system collects profile data in response to execution of the native instructions to determine execution characteristics of the non-native instruction. Thereafter, the non-native instructions and the profile statistics are fed to a binary translator operating in a background mode and which is responsive to the profile data generated by the run-time system to form a translated native image. The run-time system and the binary translator are under the control of a server process. The non-native image is executed in two different environments with first portion executed as an interpreted image and remaining portions as a translated image. The run-time system includes an interpreter which is capable of handling condition codes corresponding to the non-native architecture. A technique is also provided to jacket calls between the two execution environments and to support object based services. Preferred techniques are also provided to determine interprocedural translation units. Further, intermixed translation/optimization techniques are discussed.

62 Claims, 87 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 72

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

□ 50. Document ID: US 6044217 A

L6: Entry 50 of 63 File: USPT Mar 28, 2000

US-PAT-NO: 6044217
DOCUMENT-IDENTIFIER: US 6044217 A

TITLE: Hierarchical metadata store for an integrated development environment

DATE-ISSUED: March 28, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Brealey; Christopher L.	North York			CA
Johnston; Jeffrey G.	Scarborough			CA
Klicnik; Vladimir	Oshawa			CA
Lauzon; David M.	Etobicoke			CA
Loi; Lok T.	East York			CA
Seelemann, II; Dirk A.	Thornhill			CA

US-CL-CURRENT: 717/107; 707/104.1, 717/114

ABSTRACT:

A metadata repository for use in an integrated development environment is provided. The metadata repository is layered to define levels of common behaviour useful to different types of application development tools. The most general use tools have access to metadata at the level of simple constructed types; more specialised tools have access to components that contain properties of a target language; highly specialised tools have access to composed partitionable part metadata that can be used for constructing distributed applications.

18 Claims, 3 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 3

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Terms	Documents
L5 AND container AND proxy	63

Display Format: REV Change Format

Previous Page Next Page Go to Doc#

Refine Search

Search Results -

Terms	Documents
L6 AND JNI AND JAVA AND C++	1

Database:

- US Pre-Grant Publication Full-Text Database
- US Patents Full-Text Database
- US OCR Full-Text Database
- EPO Abstracts Database
- JPO Abstracts Database
- Derwent World Patents Index
- IBM Technical Disclosure Bulletins

Search: L7

Refine Search

Recall Text ⬍ Clear Interrupt

Search History

DATE: Sunday, January 25, 2004 Printable Copy Create Case

Set Name side by side	**Query**	**Hit Count**	**Set Name** result set
DB=USPT; PLUR=NO; OP=OR			
L7	L6 AND JNI AND JAVA AND C++	1	L7
L6	L5 AND container AND proxy	63	L6
L5	L4 AND component	2541	L5
L4	L3 OR L2	6218	L4
L3	719/$$$.ccls.	2540	L3
L2	717/$$$.ccls.	4051	L2
L1	719/310,332.ccls.	406	L1

END OF SEARCH HISTORY

Refine Search

Search Results -

Terms	Documents
L6 AND JNI AND JAVA AND C++	1

Database:

US Pre-Grant Publication Full-Text Database
[✓ US Patents Full-Text Database]
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L7

Refine Search

Recall Text

Clear

Interrupt

Search History

DATE: Sunday, January 25, 2004 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=NO; OP=OR		
L7	L6 AND JNI AND JAVA AND C++	1	L7
L6	L5 AND container AND proxy	63	L6
L5	L4 AND component	2541	L5
L4	L3 OR L2	6218	L4
L3	719/$$$.ccls.	2540	L3
L2	717/$$$.ccls.	4051	L2
L1	719/310,332.ccls.	406	L1

END OF SEARCH HISTORY

Refine Search

Search Results -

Terms	Documents
L6 AND JAVA	37

Database:

- US Pre-Grant Publication Full-Text Database
- US Patents Full-Text Database
- US OCR Full-Text Database
- EPO Abstracts Database
- JPO Abstracts Database
- Derwent World Patents Index
- IBM Technical Disclosure Bulletins

Search: L8

Refine Search

Recall Text   Clear   Interrupt

Search History

DATE: Sunday, January 25, 2004   Printable Copy   Create Case

Set Name **side by side**	**Query**	**Hit Count**	**Set Name** **result set**
	DB=USPT; PLUR=NO; OP=OR		
L8	L6 AND JAVA	37	L8
L7	L6 AND JNI AND JAVA AND C++	1	L7
L6	L5 AND container AND proxy	63	L6
L5	L4 AND component	2541	L5
L4	L3 OR L2	6218	L4
L3	719/$$$.ccls.	2540	L3
L2	717/$$$.ccls.	4051	L2
L1	719/310,332.ccls.	406	L1

END OF SEARCH HISTORY